小白成长日记——第九天

我变强了，也要秃了

逻辑训练

图形输出

1. 正方形

```
//正方形
"use strict"
var l = 4;
var f = "@";
var k = "n";
rect(l,f,k);

function rect(l,f,k){
  if(k === "y"){
    for(var i = 0; i < l; i++){
      for(var j = 0; j < l; j++){
        if(i != 0 && i != (l-1) && j != 0 && j != (l-1)){
          process.stdout.write("  ");
        }else{
          process.stdout.write(" " + f);
        }
      }
      console.log();
    }
  }else{
    for(var i = 0; i < l; i++){
      for(var j = 0; j < l; j++){

        process.stdout.write("  " + f);

      }
      console.log();
    }
  }
}
module.exports = {
  rect:rect
};
```

2. 三角形

```
//正方形
"use strict"
var l = 4;
var f = "@";
var k = "n";
rect(l,f,k);

function rect(l,f,k){
  if(k === "y"){
    for(var i = 0; i < l; i++){
      for(var j = 0; j < l; j++){
        if(i != 0 && i != (l-1) && j != 0 && j != (l-1)){
          process.stdout.write("  ");
        }else{
          process.stdout.write(" " + f);
        }
      }
      console.log();
    }
  }else{
    for(var i = 0; i < l; i++){
      for(var j = 0; j < l; j++){

        process.stdout.write("  " + f);

      }
      console.log();
    }
  }
}
module.exports = {
  rect:rect
};
```

3. 菱形

```
//菱形
"use strict"
var h = 11;
var f = "-";
var k = "n";
diamond(h,f,k);
function diamond(h,f,k){

  if(k === "y"){
    for(var i = 1; i <= ( h + 1 ) / 2; i++){
      for(var j = 1; j <= (h - i); j++){
        process.stdout.write("  ");
      }
      for(var k = 1; k <= (2 * i - 1); k++){
        if(k === 1){
          process.stdout.write(f);
        }else if(k === (2 * i -1)){
          process.stdout.write(" " + f);
        }else{
          process.stdout.write("  ");
        }
      }
      console.log();

    }
    for(var i = 1; i <= h - (h + 1) / 2; i++){
      for(var n = 1; n <= (h - 2); n++){
        process.stdout.write(" ");;

      }
      for(var j = 1; j <= i; j++){
        process.stdout.write("  ");
      }
      for(var k = h-2; k > (2*i) - 2; k--){
        if(k === h-2){
          process.stdout.write(" " + f);
        }else if(k === (2 * i -1)){
          process.stdout.write(" " + f);
        }else{
          process.stdout.write("  ");
        }
      }
      console.log();
    }

  }else{
    for(var i = 1; i <= ( h + 1 ) / 2; i++){
      for(var j = 1; j <= (h - i); j++){
        process.stdout.write("  ");
      }
      for(var k = 1; k <= (2 * i - 1); k++){
        if(k === 1){
          process.stdout.write(f);
        }else{
          process.stdout.write(" " + f);
        }
      }
      console.log();

    }
    for(var i = 1; i <= h - (h + 1) / 2; i++){
      for(var n = 1; n <= (h - 2); n++){
        process.stdout.write(" ");;

      }
      for(var j = 1; j <= i; j++){
        process.stdout.write("  ");
      }
      for(var k = h-2; k > (2*i) - 2; k--){
        process.stdout.write(" " + f);
      }
      console.log();
    }

  }
}
module.exports = {
  diamond:diamond
};
```

4. 回型

```
 //回型
"use strict";
var l = 7;
var f = "#";
var k = "y";
hui(l,f,k);

function hui(l,f,k){
  if(k === "y"){
    for(var i = 1; i <= l; i++){
      if(i === 1 || i === l){
        for(var j = 1; j <=l; j++){
          process.stdout.write(" " + f);
        }
      }else{
        for(var j = 1; j <=l; j++){
          if(j === 1 || j === l){
            process.stdout.write(" " + f);

          }else{
            if(i >= 3 && i <= (l-2)&& j >= 3 && j<= (l-2)){//内口
              if(i === (l+1)/2 &&j === (l+1)/2){
                process.stdout.write("  ");
              }else if(i >= 4 && i <= (l-3)&& j >= 4 && j<= (l-3)){
                process.stdout.write("  ");
              }else{

                process.stdout.write(" " +f);
              }

            }else{
              process.stdout.write("  ");
            }
          }
        }
```

```
            }
            console.log();
        }

    }else{
        for(var i = 1; i <= l; i++){
            if(i === 1 || i === l){
                for(var j = 1; j <=l; j++){
                    process.stdout.write(" " + f);
                }
            }else{
                for(var j = 1; j <=l; j++){
                    if(j === 1 || j === l){
                        process.stdout.write(" " + f);

                    }else{
                        if(i >= 3 && i <= (l-2)&& j >= 3 && j<= (l-2)){//内口
                            if(i === (l+1)/2 &&j === (l+1)/2){
                                process.stdout.write("  ");
                            }else{
                                process.stdout.write(" " + f);
                            }
                        }else{
                            process.stdout.write(" " +f);
                        }
                    }
                }
            }
            console.log();
        }

    }
}
module.exports = {
    hui:hui
};
```

5. 梯形

```
//梯形
var s = 3;
var h = 5;
var f = "$";
var k = "y";
ti(s,h,f,k);

function ti(s,h,f,k){
    if(k === "y"){

        for(var i = 1; i <= h;i++){
            for(var j = 1 ; j<= s; j++){
                if(i===1||i===h){
                    process.stdout.write(" " + f);
                }else if(j === 1|| j === s){
                    process.stdout.write(" " + f);
                }else{
                    process.stdout.write("  ");
                }
            }
            s++;
            console.log();
        }

    } else{
        for(var i = 1; i <= h;i++){
            for(var j = 1 ; j<= s; j++){
                process.stdout.write(" " + f);
            }
            s++;
            console.log();
        }
    }
}
module.exports={
    ti:ti
};
```

字符串

1. 字符串长度

```
"use strict";
//字符串长度
var s = "";
var count = 0;

l(s);

function l(n){
    for(var i in n){
        count++;
    }
    console.log(count);
}
```

2. 分割字符串

```
"use strict";
//分割字符串
var str = "abcdcfg";
var f = "cd"
var x = Array();
var y = 0;
var z = '';


sp(str, f);


function sp( str, f){
    var count1 = 0;
    var count2 = 0;

    for(var i in str){
        count2+=1;
    }

    for(var s in f){
        count1+=1;
    }

    for(var i = 0;i < count2; i++){
        var q ='';
        for(var j = 0; j < count1; j++){
            q += str[i+j]
        }
        if(q === f){
            x[y] = z;
            y += 1;
            z = '';
            i += count1 - 1;
        }else{
            z += str[i];
        }

    }
    x[y] = z;
    console.log(x);
}
```

3. 字符串合并

```
"use strict";

var a =["ab"];
var b = "-";

he(a,b);

function he(a,b){
    var count = 0;
    var c = "";

    for(var i in a){
        count++;
    }
    for(var i = 0; i < count; i++){
        if(i != count - 1){
            c += a[i] + b;
        }else if(i === (count-1)){
~/文档/小白成长日记/小白成长日记——第九天.md
GitHubGit (0)1 update
Uncaught TypeError: Right-hand side of 'instanceof' is not callable
/home/jinjiaxin/.atom/packages/markdown-scroll-sync/lib/main.coffee:38
Show Stack Trace
The error was thrown from the markdown-scroll-sync package. Atom is out
of date: 1.38.2 installed; 1.39.1 latest. Upgrading to the latest version may fix
this issue.
            c += a[i];
        }



    }
    console.log(c);
}
```

4. 字符串搜索

```
"use strict";
//字符串搜索
~/文档/小白成长日记/小白成长日记——第九天.md
GitHubGit (0)1 update
Uncaught TypeError: Right-hand side of 'instanceof' is not callable
/home/jinjiaxin/.atom/packages/markdown-scroll-sync/lib/main.coffee:38
Show Stack Trace
The error was thrown from the markdown-scroll-sync package. Atom is out
of date: 1.38.2 installed; 1.39.1 latest. Upgrading to the latest version may fix
this issue.
var str = "abcdefg";
var f = "acdcbc";
var x = new Array();
var y = 0;
var h = 0;

fen(str,f)


function fen(str,f){
    var count1 = 0;
    var count2 = 0;
    for(var i in str){
        count1++;
    }
    for(var s in f){
        count2++;
    }



    for(var i = 0; i<count1; i++){
        var a = "";
        for(var j = 0; j < count2; j++){
            a += str[i+j];
            if(a === f){
                console.log(i);
                return;

            }else if(i === count1 - 1 && a!=f){
                console.log(-1);
                return;
            }
        }

    }

}
```

5. 字符串截取

```
"use strict";
//截取字符串
var str = "abcdcfg";
var begin =-4;
var l =44;

sp(str, begin, l);

function sp( str, b, l){
    var count2 = 0;


    for(var i in str){
        count2+=1;
    }



    var q ='';
    for(var i = b;i < b+l; i++){
```

```
    q += str[i];
   if(b < 0 || b > count2){
      console.log("起始位置不合理！！");
      return;
   }else if((b + l) >count2){

    console.log("超出字符串长度，无法截取！");
    return;
   }
  }
  console.log(q);

}
```